Ship Smart System Design:

A simulation based design environment for ship systems

Presented at
Center for Innovation in Ship Design
Navsea - ONR - NSWC/CD
8 July 2004

Dr. Roger Dougal

Dept of Electrical Engineering
University of South Carolina
Columbia, SC 29208

Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

Outline

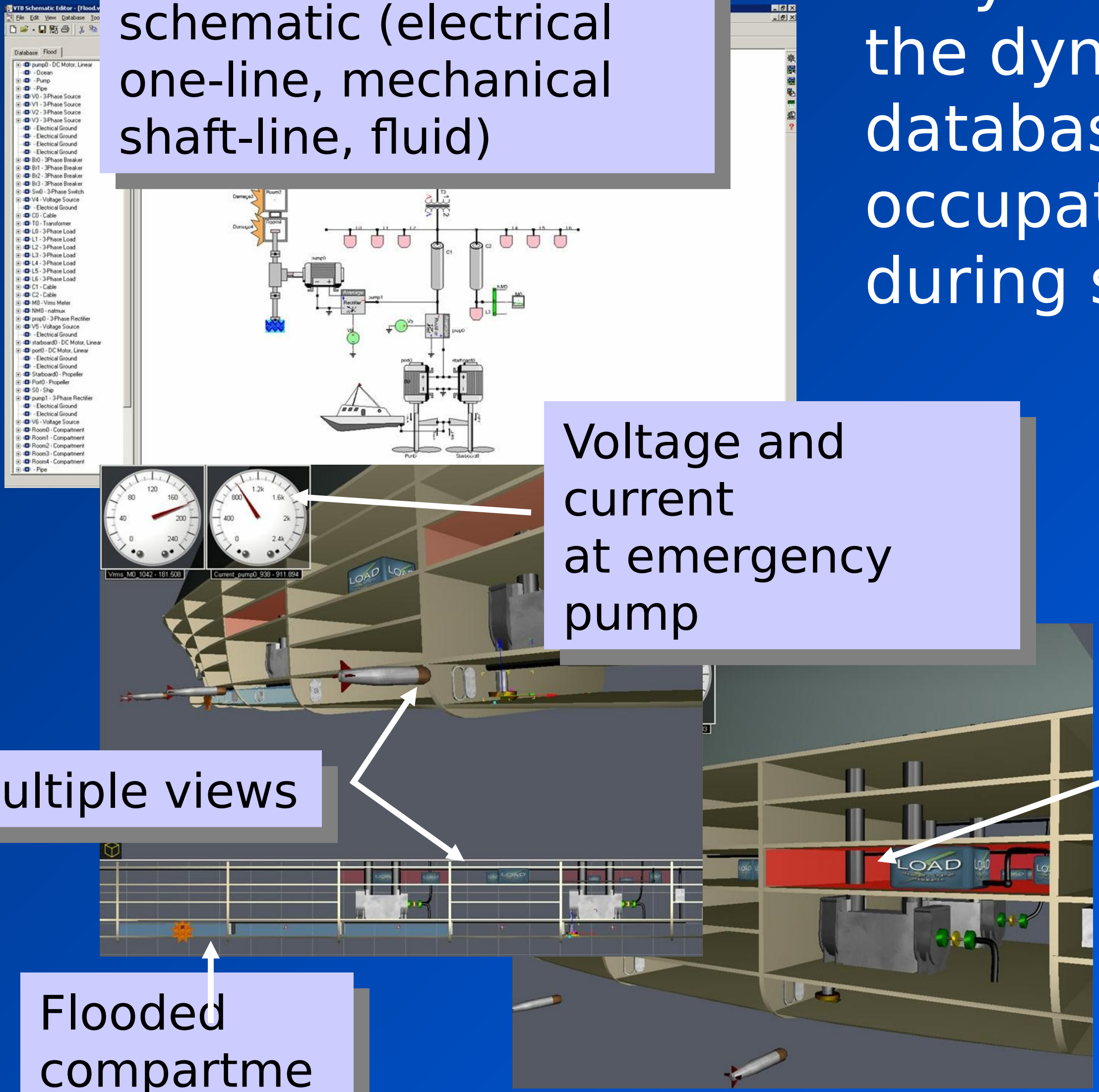
- Objectives
- Participants
- History
- Concepts
- Approach
- Status

S3D

- S3D will take the technologies developed under the VTB project and add new capabilities that specifically support the ship life-cycle processes, beginning with design

- The focus application of the first user tool is the military shipbuilding industry

Fully interactive views into the dynamic system database from many occupational perspectives during simulation runtime



Dynamic system schematic (electrical one-line, mechanical shaft-line, fluid)

Voltage and current at emergency pump

Multiple views

Flooded compartment

Structure info directly from CAD

Overheated compartment

Changes in 3D view immediately reveals effects in simulation

S3D Concepts

- Comprehensive ship systems tool, useful throughout the life-cycle, from design phase to operations

Design	**Operations**
Contract	Test, Trials, Operations
Detail	Crew Training
	Ship Alterations

- Many views into the ship systems, based on the same database and underlying simulation models
  - Discipline specific views
  - Task specific views
- First emphasis will be on the *design task* for
  - Electric systems
  - Fluid systems

The CAD view of a ship

- A ship is a complex multi-disciplinary system in which many interactions occur
- Discipline-specific CADs each present one narrow, highly-filtered, non-interacting view of that complex, multi-disciplinary system
- An integrated CAD environment is required to develop a true multi-disciplinary description of a ship that can predict the operating characteristics of the real system

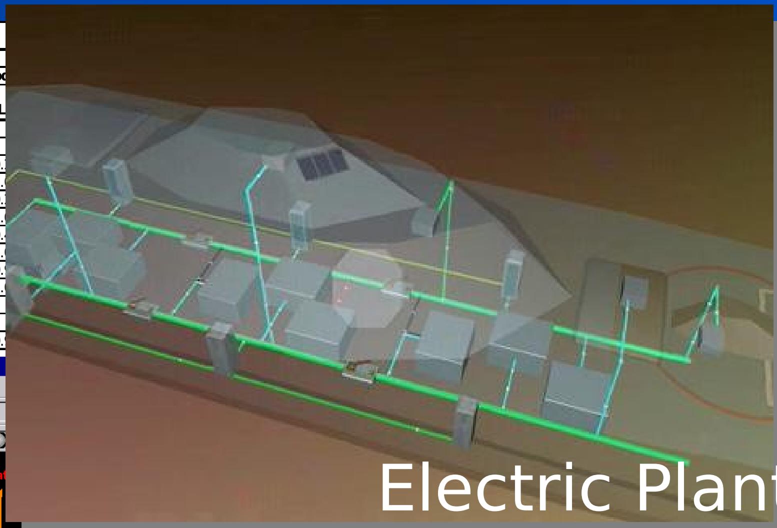
Discipline-specific (filtered) views of a single database

Nav architect

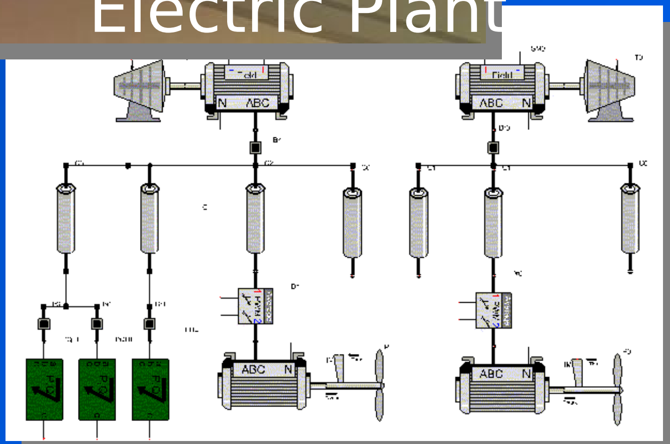
PROPULSION MACHINERY

ITEM	DESCRIPTION OF EQUIPMENT	RATE	QTY	VOLT	EST BR	NP	EFF	KW PER UNIT	TOT CON KW	SEA 60Hz Port SWBD #	LF	KW	Stbd #	L
1	BOW THRUSTER		1	5500	2200	2237	0.96	2282	2282					
2	STEERING GEAR PUMP		4	460	65.0	97	0.92	70.7	282.6	1	0.20	14.1	1	0.
3	CENTRAL F W COOLING PUMP		6	460	48.0	65	0.94	51.1	306.4	2	1.00	102.1	2	1.
4	STARTING AIR COMPRESSOR		4	440	20.0	22	0.90	22.2	88.9	1	0.20	4.4	1	0.
5	CENTRAL S W COOLING PUMP		6	460	36.0	42	0.92	39.1	234.8	2	1.00	78.3	2	1.
6	CPP HYDR PWR UNIT		4	440	20.0	22	0.91	22.0	87.9	1	0.40	8.8	1	0.
7	MAIN ENG L O PUMP		4	460	57.0	64	0.94	60.6	242.6	1	1.00	60.6	1	1.
8	F O PURIFIER		4	460	9.6	14.3	0.88	10.9	43.6	1	0.90	9.8	1	0.
9	MAIN ENG J W COOLING PUMP		4	460	11.8	18	0.90	13.1	52.4	1	1.00	13.1	1	1.
10														
11	CPP PRESSURE MAINTAINING PUMP		2	440	1.2	1.3	0.78	1.5	3.1					
12	MAIN ENG L O PURIFIER		2	460	2.2	5.9	0.87	2.5	5.1	1	0.90	2.3	1	0.
13	BOW THRUSTER HYD PO		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]					
14	MAIN ENG FUEL CIRC PU													
15	STERN TUBE L O COALES													
16	MAIN ENG EXH VLV L O B													
17	MAIN ENG F O SUPPLY PU													
18														
19	F O VISCOSITY CONTROL													
20	MAIN ENG J W PREHEATE													

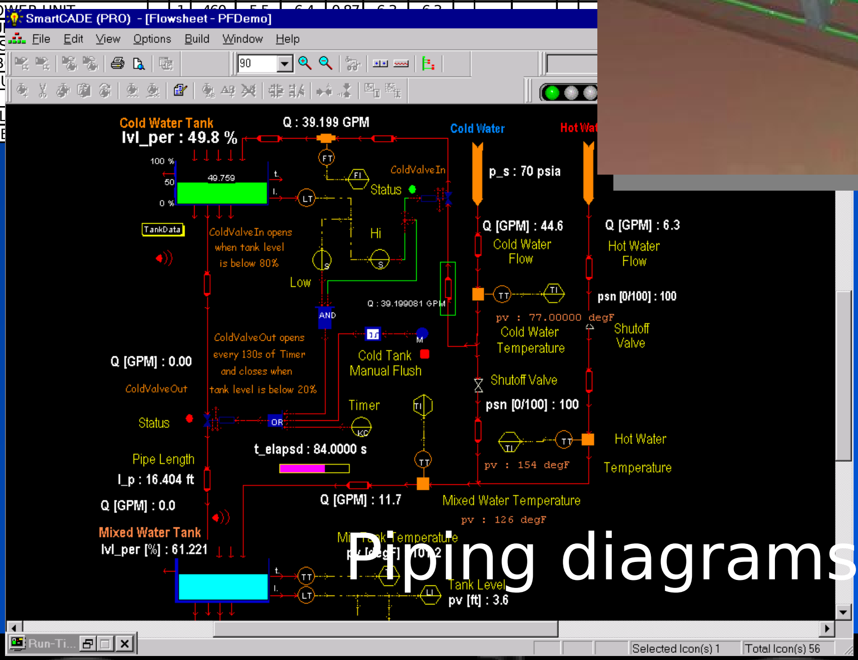
Structures



Electric Plant



Piping diagrams



Advanced Simulation Concepts

- Current user profile determines which filter is applied to produce the current system view
  - Power engineer - one-line diagram
  - Electronics engineer - circuit diagrams
  - Piping engineer - piping diagram
  - Naval architect - physical size, mass, location
- View-centric simulation
  - Level of detail consistent with user focus
    - Time resolution
    - Accuracy or physics resolution
- User may interact with the system through any schematic view or physical view
- Accomodate uncertainty

Conceptual Electric System Design Process

- Begin from standard parts list
- Use spreadsheet to list sources and loads and perform preliminary load analysis
- Semi-automatically allocate loads onto one or more “standard” power system topologies
- Analyze and adjust/optimize system performance
- Verify requirements met

Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

S3D Participants

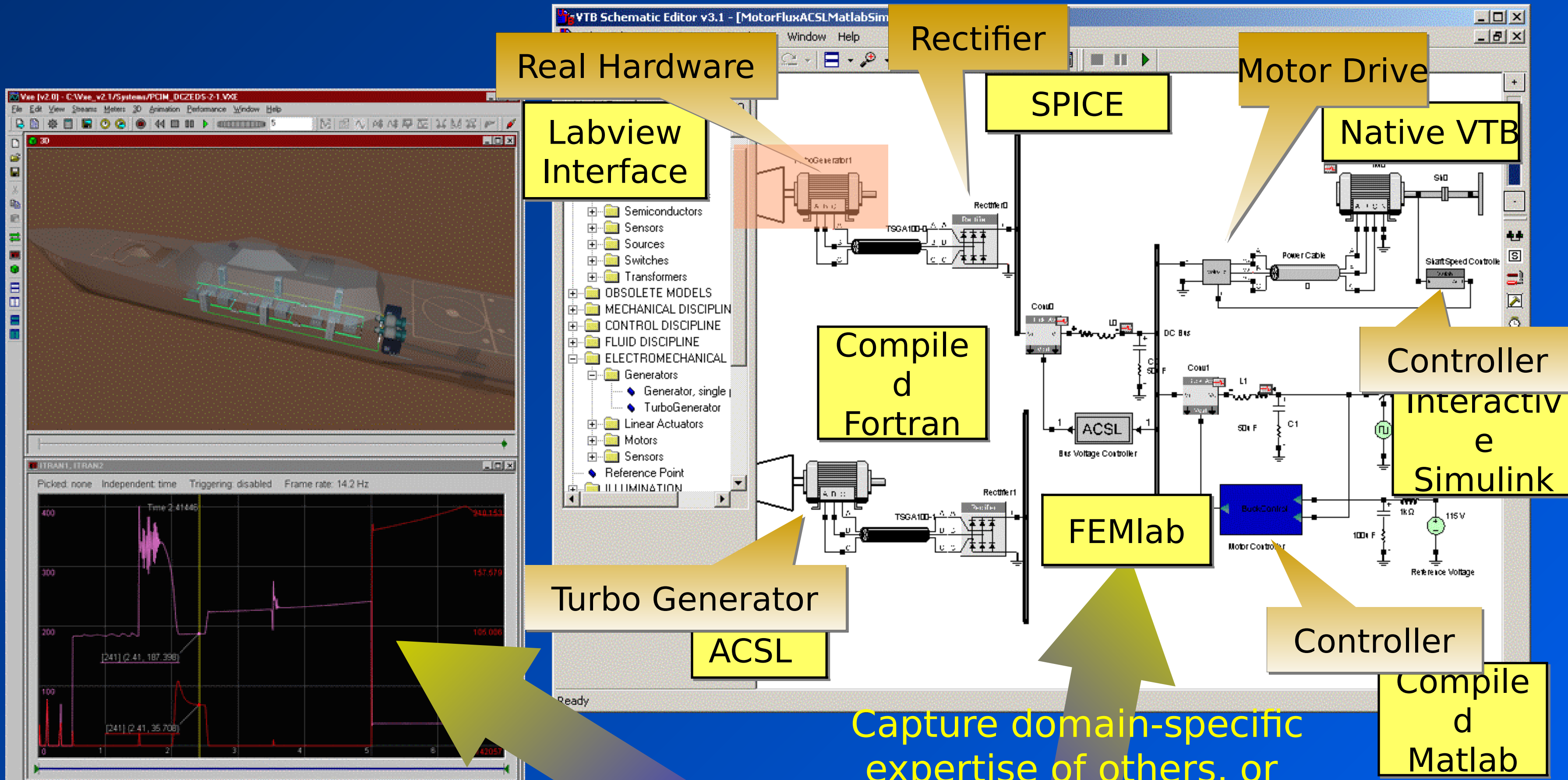
S3D Direct Participation
VTB-related Partners

Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

VTB Goals

- Develop the worlds most capable, flexible, applicable, simulator for interdisciplinary dynamic systems
  - Provide an interactive, immersive, realtime, adaptable simulation environment that is supported on many platforms
  - Support top-down design approaches, accounting for parameter uncertainty, partial data, complex multiresolutional models, many model tools
  - Support incremental virtual prototyping via hardware in the simulation loop



Advanced visualizations increase comprehension

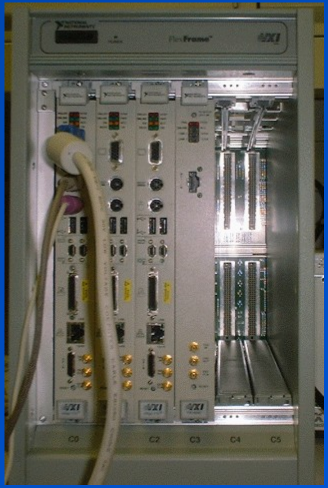
Capture domain-specific expertise of others, or preserve utility of existing models and modeling skills

7 years of development has yielded a very capable tool

VTB is now a Pervasive Computing Environment

Notebook PC

Desktop PC
- MS Windows
- Linux

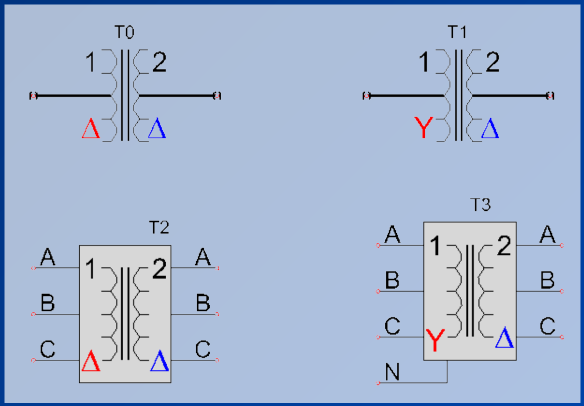


VXI bus industrial controllers

High performance Quad Intel Itanium

Simulations can be
- User interactive (immersive)
- Hardware interactive (hardware in the loop)

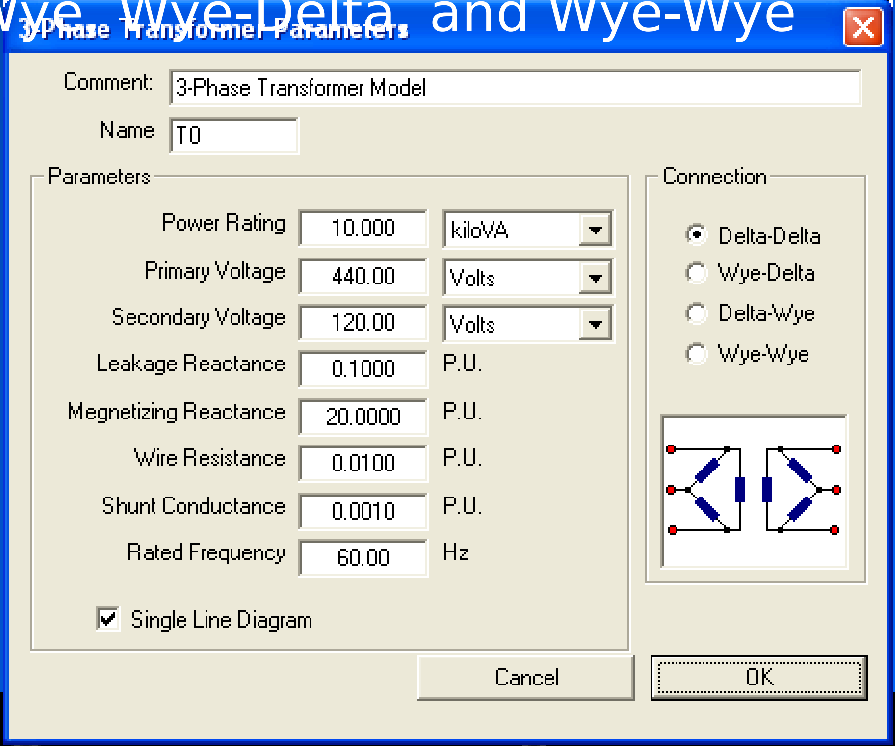
VTB Model Example: Transformer



Features:

• Model accounts for leakage reactance, finite magnetizing reactance, copper losses

• Supported configurations: Delta-Delta, Delta-Wye, Wye-Delta, and Wye-Wye

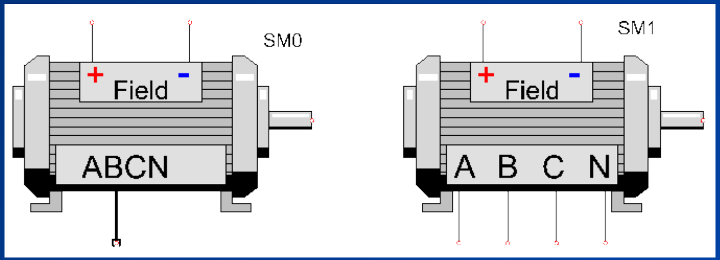


• Multi-layer icon is employed

• Single line diagram drawing is supported

• Model is compatible with other 3-phase models in terms of terminal connections and single line diagram drawing

VTB Model Example: Synchronous Machine



$$i_{abc}(t) = G_{abc} v_{abc}(t) - G_{abc}\Gamma v_n(t) - G_{abc}\frac{d\lambda_{abc}(t)}{dt}$$

$$i_n(t) = -\Gamma^T G_{abc} v_{abc}(t) + \Gamma^T G_{abc}\Gamma v_n(t) + \Gamma^T G_{abc}\frac{d\lambda_{abc}(t)}{dt}$$

$$i_f(t) = g_f v_f(t) - g_f\frac{d\lambda_f(t)}{dt}$$

$$T_m(t) = J\frac{d\omega_m(t)}{dt} - (i_{abcs}(t))^T\left[\frac{\partial L_{sr}(\theta_m(t))}{\partial\theta_m}\right] i_{fDQ}(t) - \frac{1}{2}(i_{abcs}(t))^T\left[\frac{\partial L_{ss}(\theta_m(t))}{\partial\theta_m}\right] i_{abcs}(t)$$

$$0 = \frac{d\theta_m(t)}{dt} - \omega_m(t)$$

$$0 = -\theta_m(t) + \omega_{sm} t + \frac{2}{p}\delta(t) + \frac{\pi}{p}$$

$$0 = \lambda_{abc}(t) - L_{ss}(\theta_m(t)) i_{abc}(t) - L_{sr}(\theta_m(t)) i_{fDQ}(t)$$

$$0 = \lambda_{fDQ}(t) - L_{rs}(\theta_m(t)) i_{abc}(t) - L_{rr} i_{fDQ}(t)$$

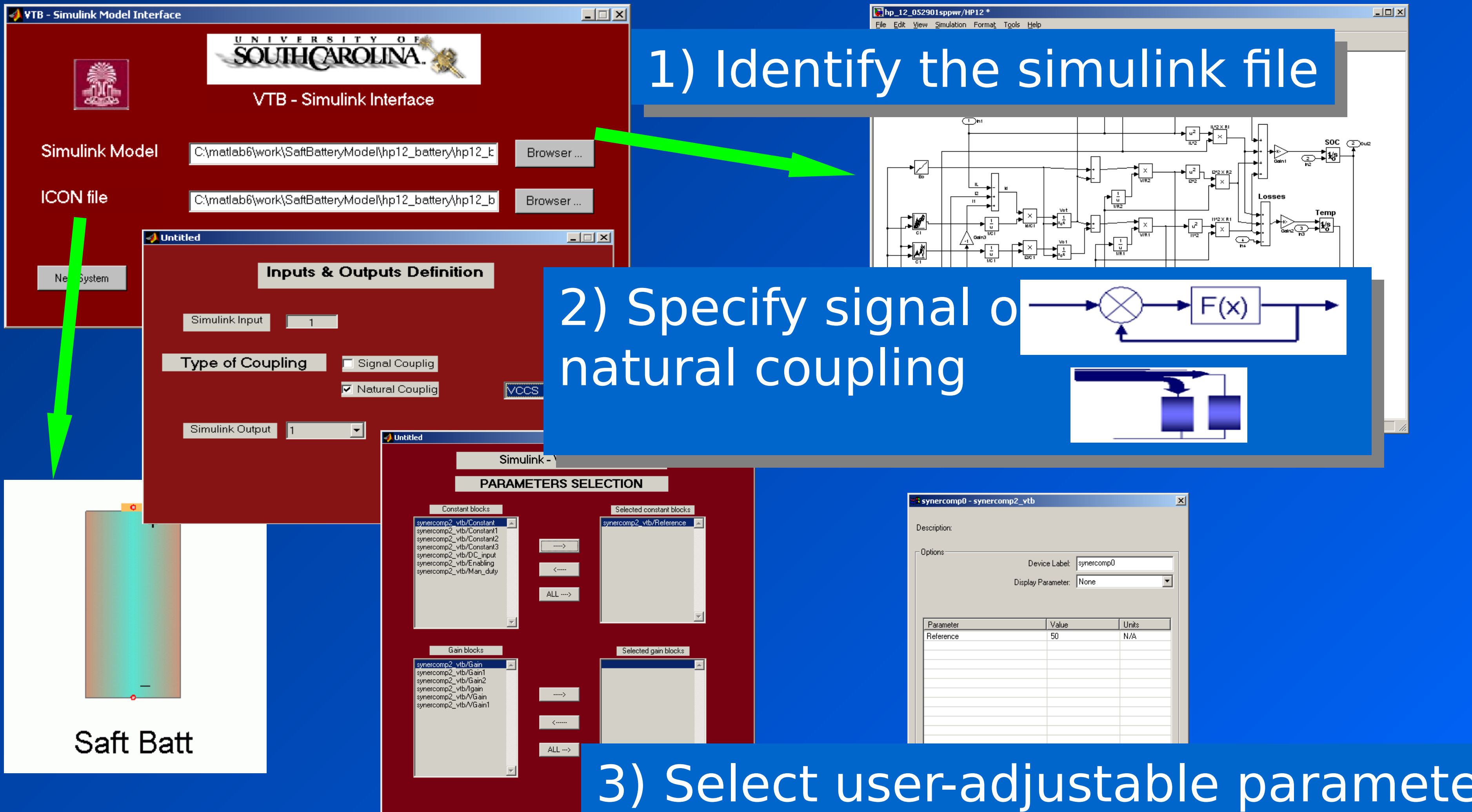
$$0 = R_{DQ} i_{DQ}(t) + \frac{d\lambda_{DQ}(t)}{dt}$$

Phase Domain Model

- Multi-layer icon is employed
- Single line diagram drawing is supported
- Model is compatible with other 3-phase models in terms of terminal connections and single line diagram drawing

- Number of Equations: 43
- Jacobian Matrix: 1849 elements
- Manual development and validation of the model: ~6 months
- Model development and validation using UDD: 2 weeks

Fast wrapping of non-native models



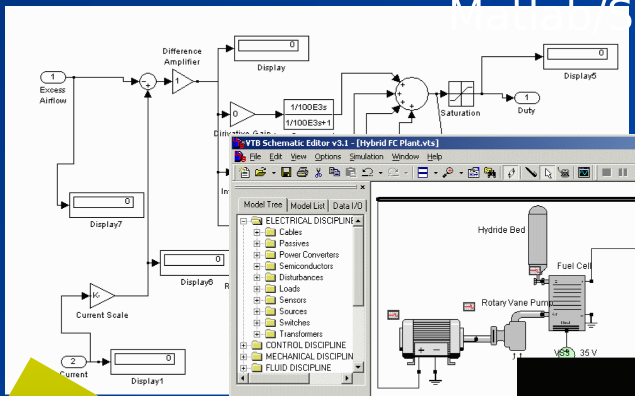
1) Identify the simulink file

2) Specify signal or natural coupling

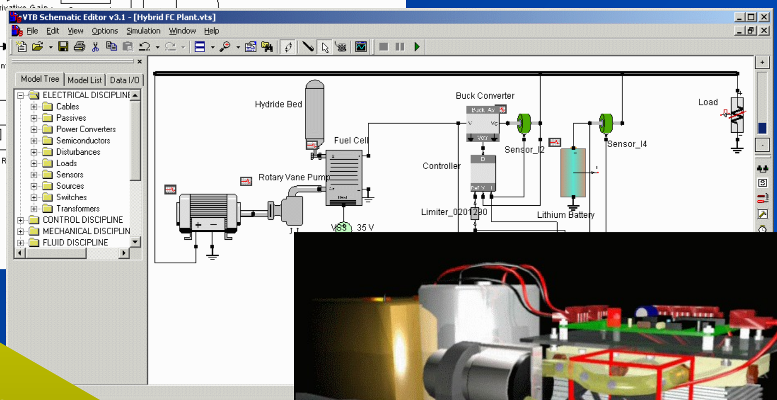
3) Select user-adjustable paramete

VTB provides a mature path for control development/insertion

Controllers are designed using Matlab/Simulink

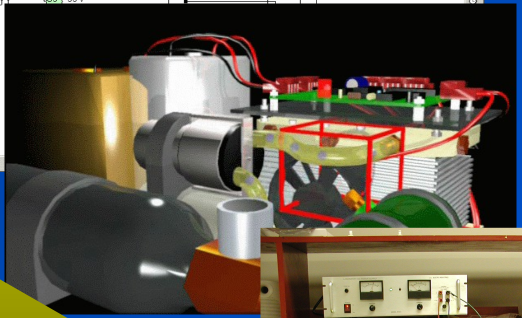


The Simulink controller is dropped into VTB…



… then interactively tested and tuned using a high-accuracy system model

After compiling, the controller model can be distributed to any partner…



…who can test the power source to see how it will work with another load

…. or the controller can be loaded onto dSpace hardware

…to test a hardware prototype

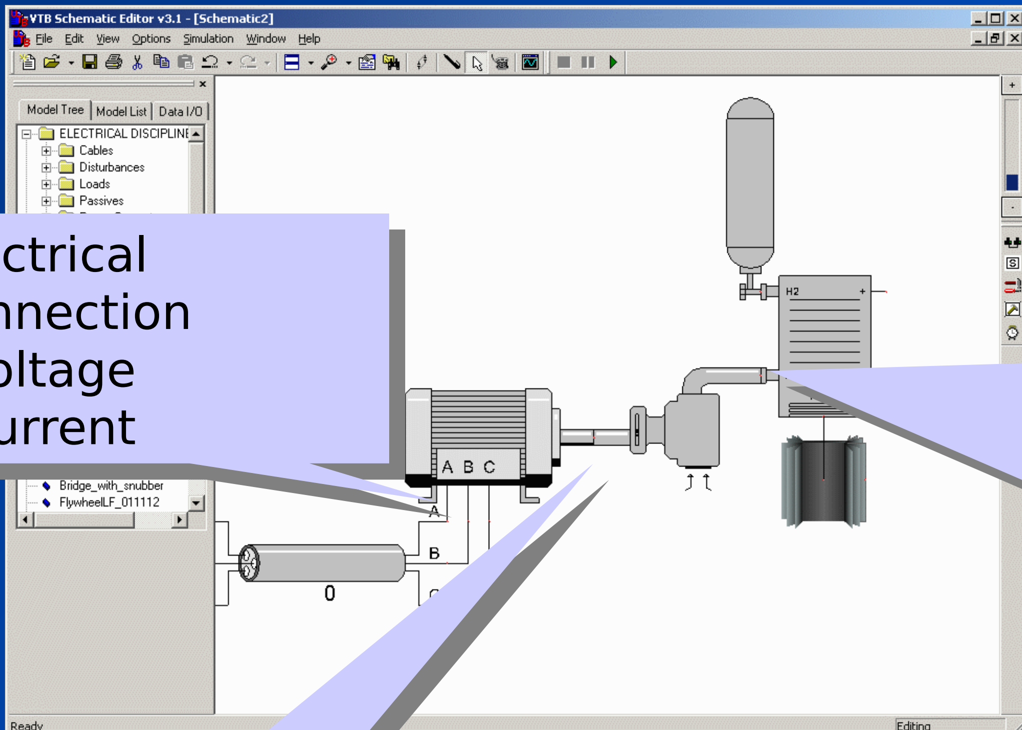
Finally test code on embedded processor that communicates with the simulation model

From concept to hardware

Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

Portable, re-useable models automatically enforce conservation laws



Electrical connection
- Voltage
- Current

Fluid connection

Pressure

O_2, N_2 , CO_2

heat

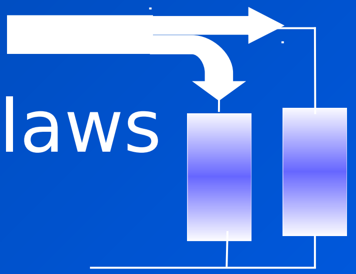
Mass flow

Shaft connection
- Torque
- Rotational Speed

VTB supports three forms of coupling between models

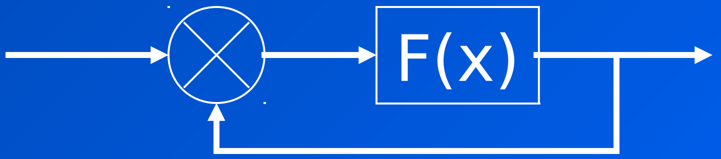
Natural Coupling

Enforces physical conservation laws



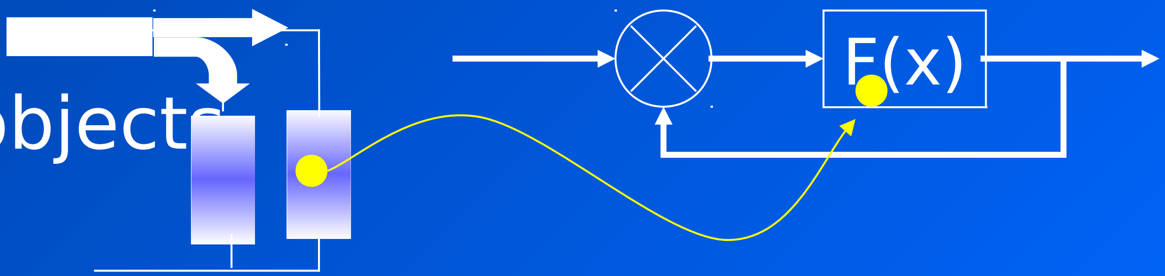
Signal Coupling

Directed flow of information through objects

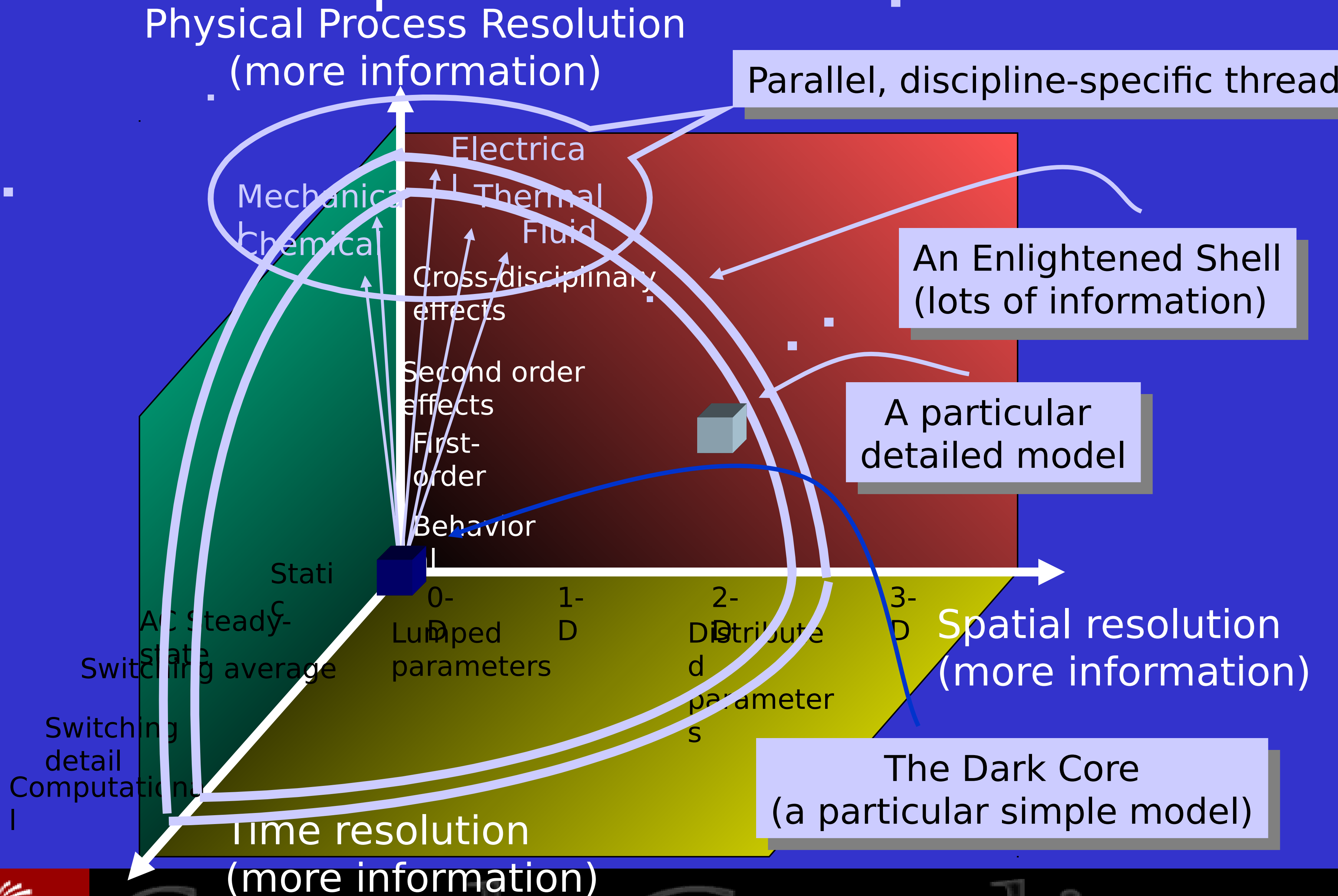


Data Coupling

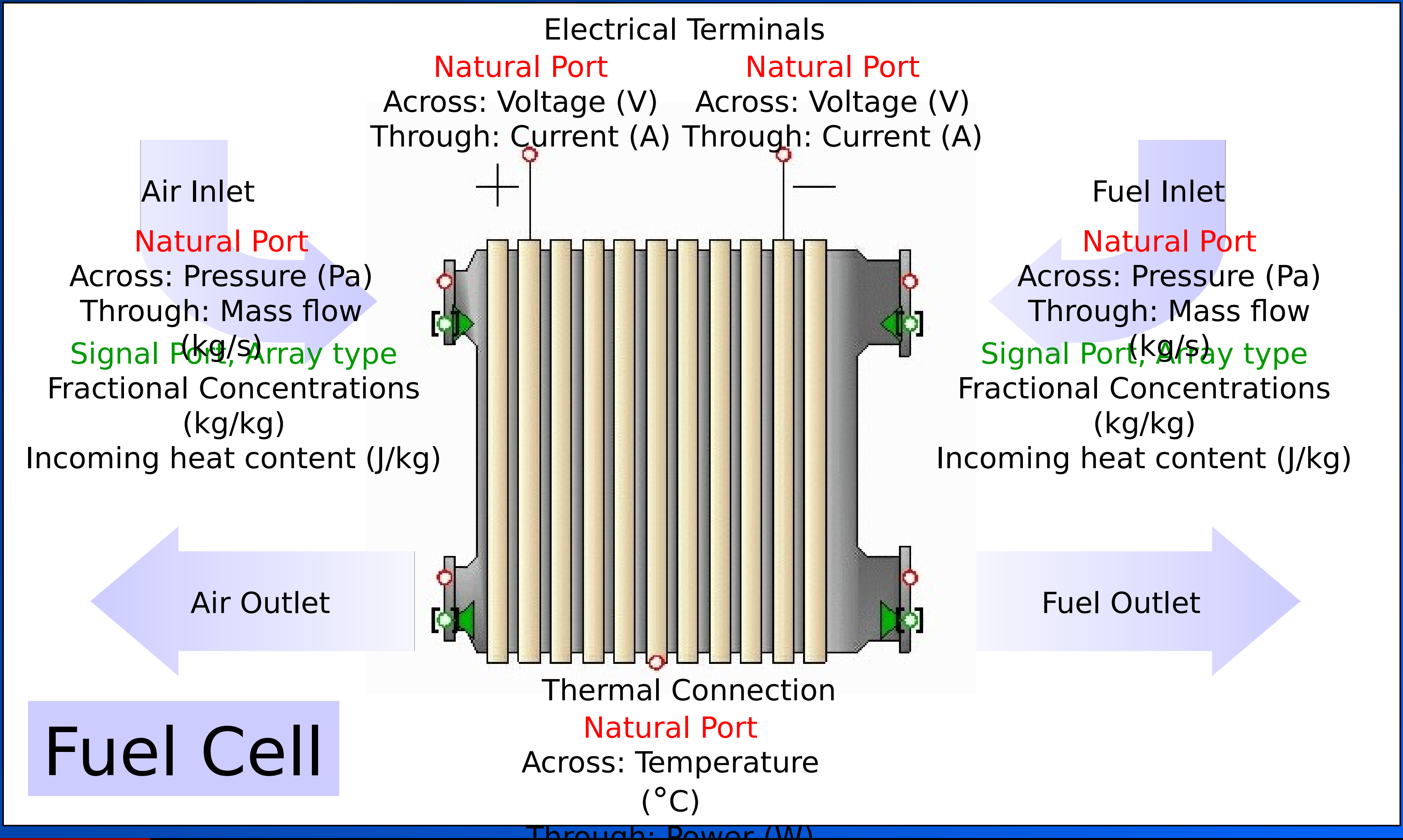
Pass data between objects



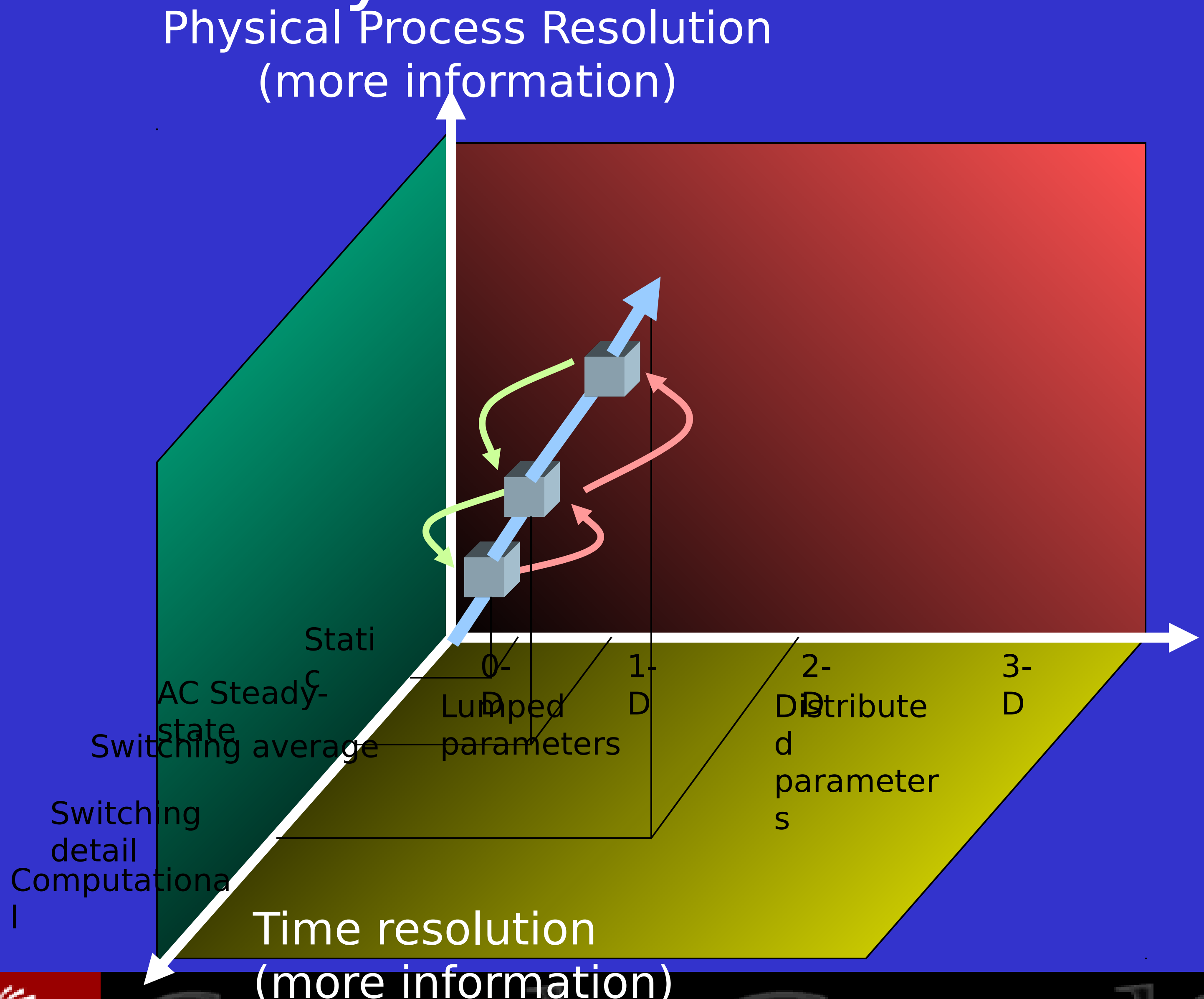
Concepts of model resolution



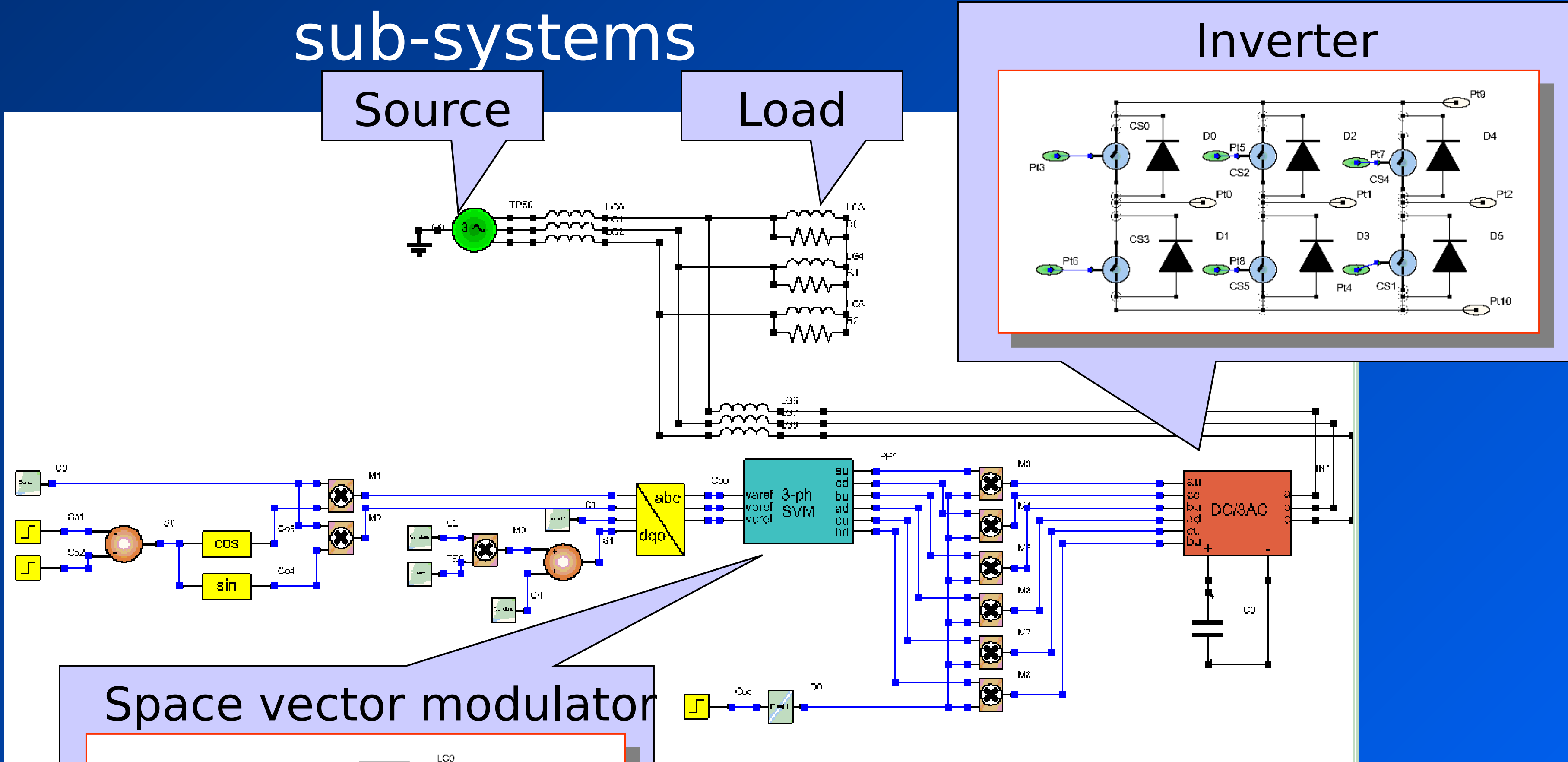
Multidisciplinary models

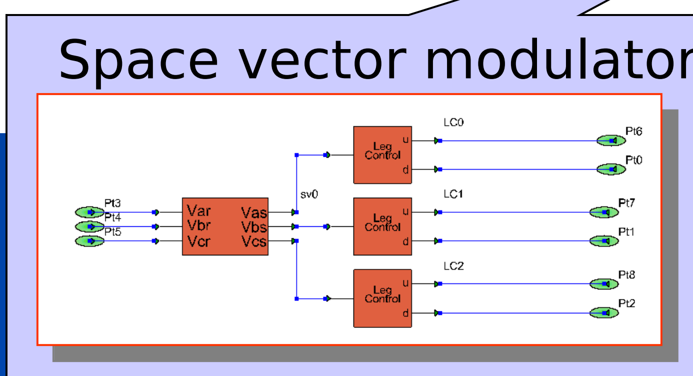


Dynamic Model Order



Hierarchical definition of sub-systems





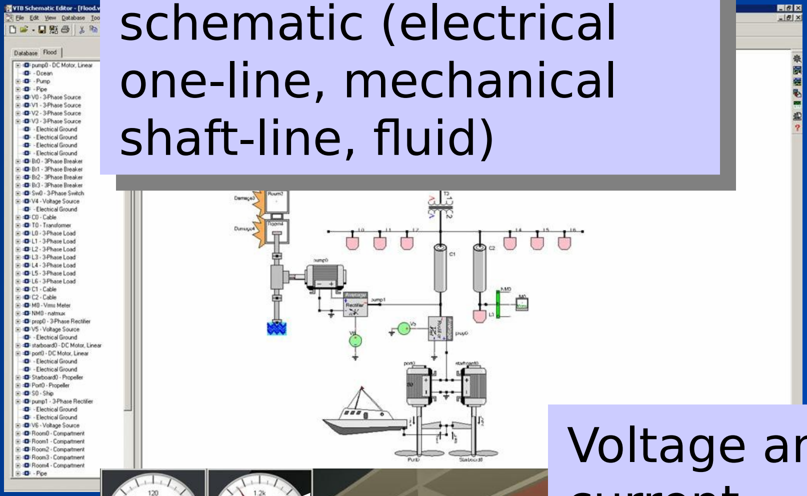
This example illustrates:
Hierarchical groupings of components
Mixed signal and natural ports

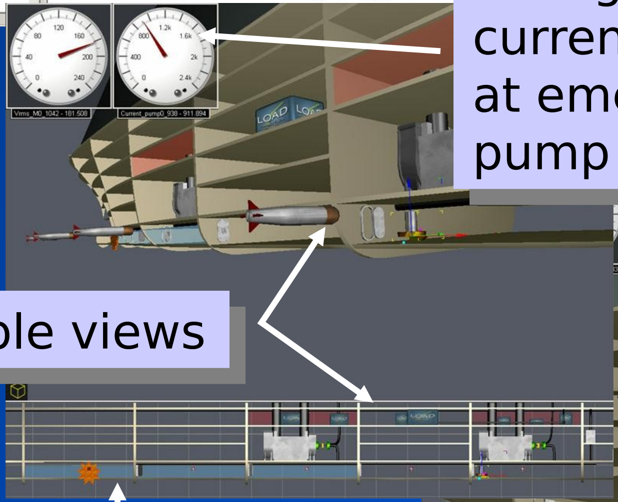
Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

Fully interactive views into the dynamic system database from many occupational perspectives during simulation runtime

Dynamic system schematic (electrical one-line, mechanical shaft-line, fluid)





Voltage and current at emergency pump

Multiple views

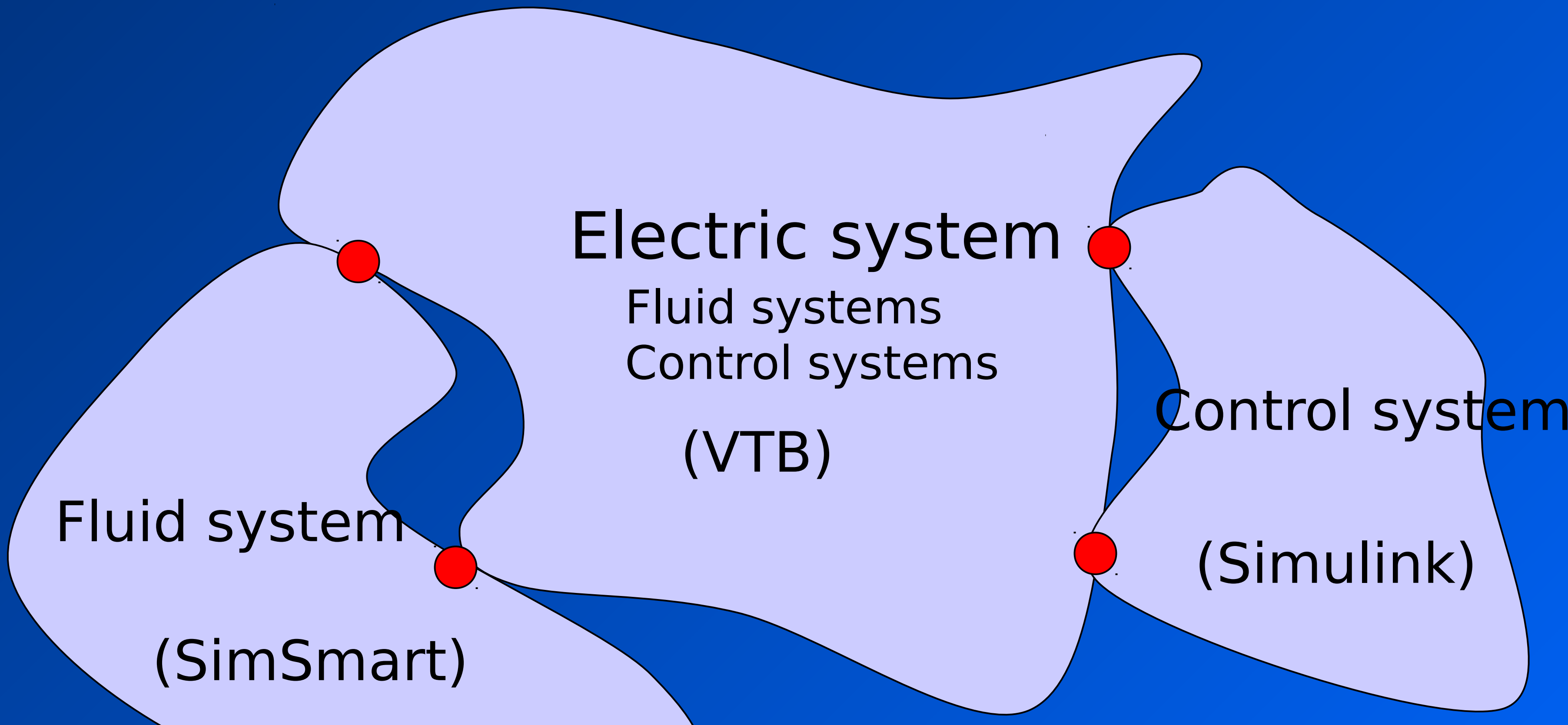
Flooded compartment

Structure info directly from CAD

Overheated compartment

Changes in 3D view immediately reveals effects in simulation

S3D Year 1 Demo



Co-simulation best supports analysis of systems with few points of tangency

Principal Tasks

SimSmart

- Develop stepping API
- Port some fluid models to VTB
- Prepare fluid subsystem model in SS (based on LPD)
- Work w/USC to resolve any coupling problems

USC

- Upgrade parts library to support S3D demonstration features
- Develop new models needed for demonstration system
- Interface to SS stepping API
- Support tech transfer to IDV
- Define new structure of model database
- Define methods for discipline-specific filters
- Define methods for view-centric simulation
- Assemble/test/troubleshoot integrated demo system model

IDV

- Modify Schematic editor
- Provide visualization models
- Modify vizor to support the S3D vision
- Assist definition of demo
- Create/maintain software development plan
- Prepare demonstration video

Architectural Differences Between VTB and S3D

XML

- In VTB, only the system definition is stored as XML
- In S3D:
- Dynamic models will be defined using XML to allow greater flexibility to extend the interface.
- Visual models will be defined in XML
- Component connectivity is defined in XML
- Mathematical description will eventually (but not currently) be defined in XML (currently still using compiled code)
  - Modeler decides if the IP rights for the component's behavior should be protected or shared
- Models can be easily shared among the S3D community

Component Technology

- The use of component technology provides greater flexibility for model developers
- Language neutral (not tied to C++, not tied to a compiler, better versioning control)
- Remotability (distributed systems)
- Rich set of services are provided as part of the framework (asynchronous messages, message queues, transaction support, web services, etc…)
- COM, COM+ Services, .NET

Separation of Model, Visualization, and Solver

- The system has been separated into multiple tiers
  - Model Framework
  - Solver Framework
  - Studio (Schematic Editor)
  - 2-D and 3-D visualization
  - Data access
- This allows for a system that can eventually be distributed and load balanced for optimal performance
- Allows for use of Web Services
- Simulations can be batched for processing off-line during non-peak hours
- Simulations can be outsourced to specialized processing hardware and farms

Framework provides more functionality

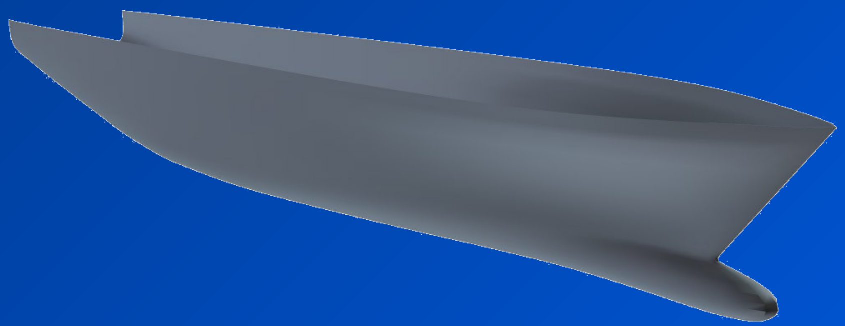
- The XML meta data of a model is interpreted by the model framework code
  - Model developers do not need to code as much since the framework provides more services
  - Eventually it would be possible to provide no code just the meta data
  - Framework allows for extensibility (e.g. to incorporate custom code if necessary)

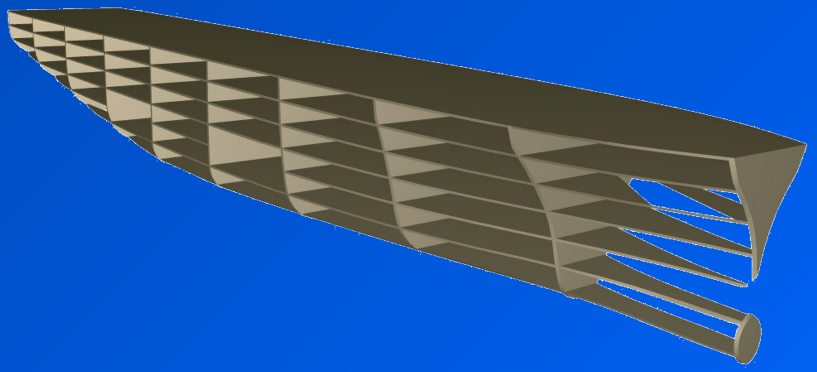
Database Support

- Framework provides support for data persistence
- Allows extensibility for supporting other business processes
- Allows better management of modeling process
  - Model promotion
  - Rollback model changes
  - Versioning models

Hull Geometry

- Hull geometry can be imported directly into the visualization enivronment using translator utilities
- Mechanical systems can be placed in 3D view and coupled to simulation models
- System simulations drive actions and model attributes in visualization

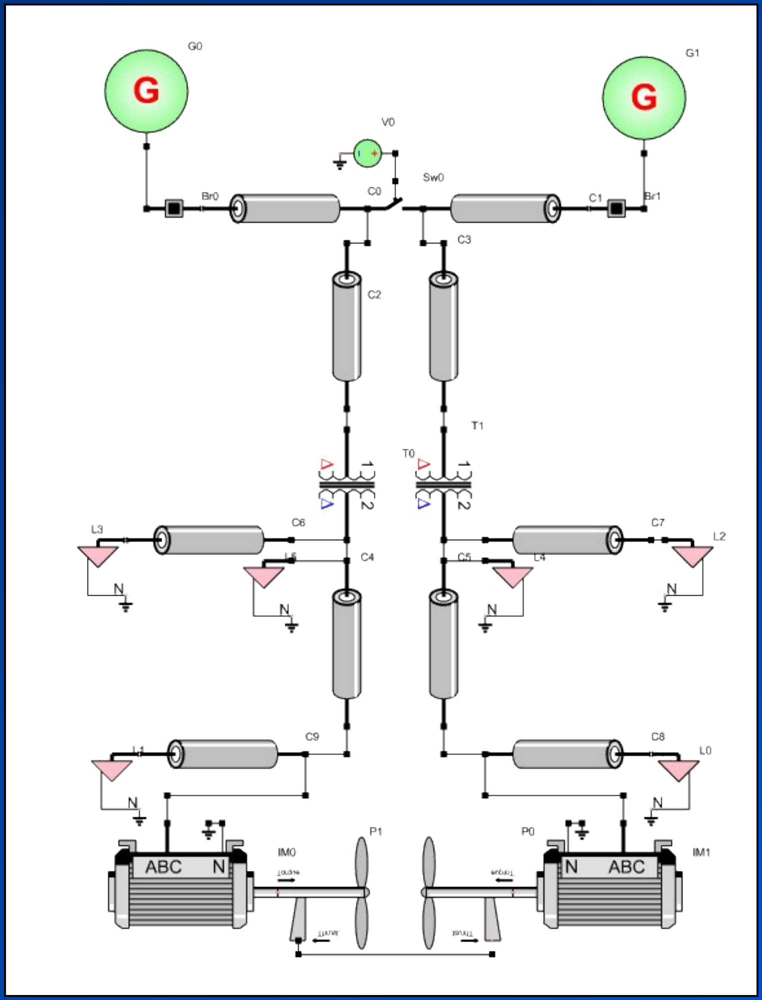




Simulated/Physical Coupling

- Visualization environment is tightly coupled with dynamic models in any number of ways:
  - Size and length (e.g. cables)
  - Direct commands built into visualization plugins
  - Identifying correspondence between entities in schematic and physical location is facilitiated by highlighting in both views
  - Spatial proximity affects simulation variables (future thermal and field effects)

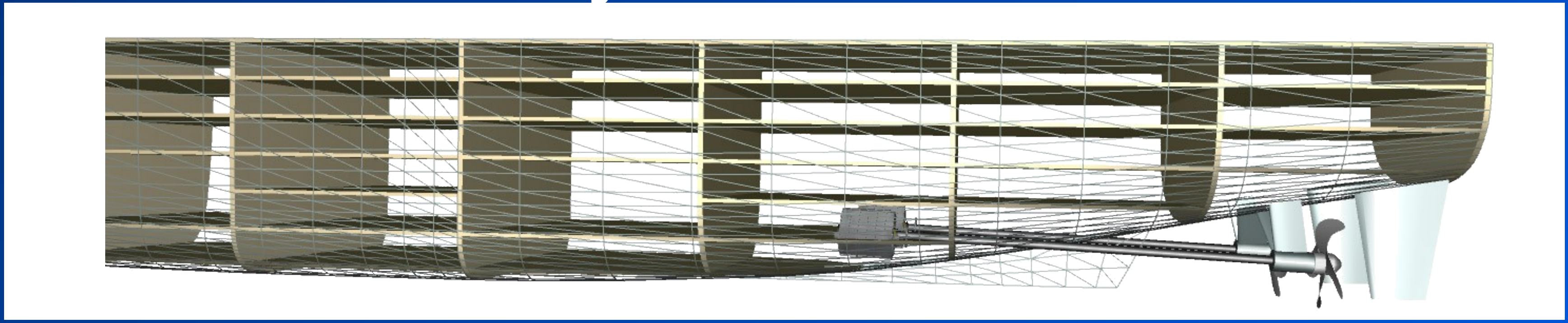
Schematic View



- Each icon corresponds to a VTB simulation model
- Models can be pulled from an existing database, or created using numerous tools
- Model attributes can be flagged and connected to the visualization/physical view as the simulation runs
- Model attributes can also be tagged as “variables” which are tied to attributes of the physical simulation
- In S3D, these couplings will be handled automatically
- Hierarchies are also supported, hiding or revealing additional detail as needed

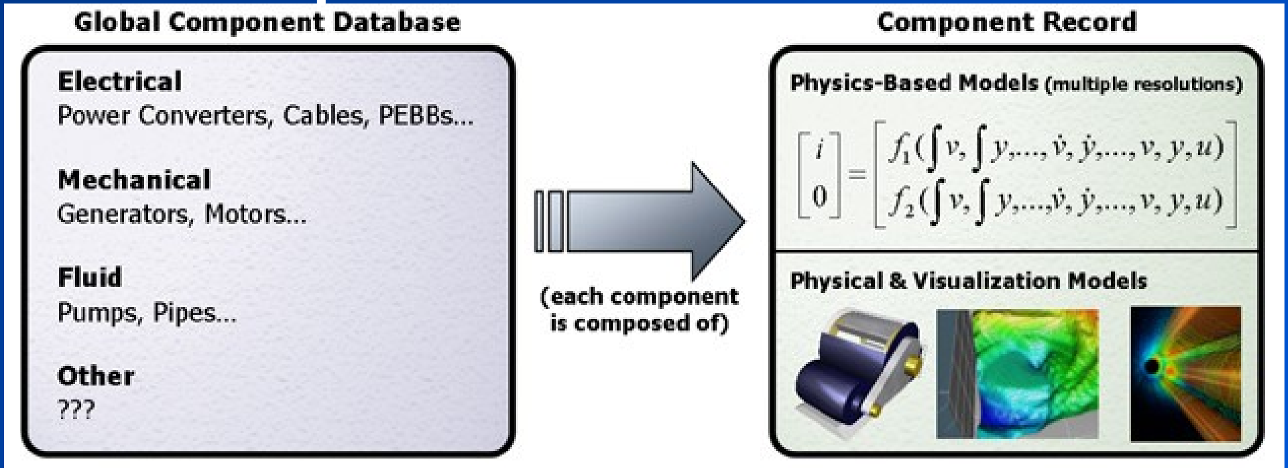
UNIVERSITY OF SOUTH CAROLINA

Physical View



- 3D components are instantiations of VXE plugins
- As with VTB, there is a standard library of existing plugins as well as the capacity to make new plugins (VXE SDK)
- Utility plugins (like 3D model loader which can load geometry from a variety of sources) can be used to build a majority of systems, without the need for customization
- 3D plug-ins do not necessarily have to represent something physical
- Multiple simultaneous views are possible
- Plugin nodes are arranged in a scene graph hierarchy

Component Database



- S3D models will be a coupling of
  - Physics-based mathematical descriptions in the form of VTB models
  - Visualization models in the form of VXE/Vizor plugins
- A component manager will facilitate the couplings between sim and viz models that are done manually now
- Models will be categorized by discipline
- Multiple resolutions of the physics-based models can be defined for each component

Software Engineering Challenges

- S3D is a software engineering challenge equally as much as it is a simulation/visualization challenge
- Software will be overhauled for commercial use.
  - Uniform standards will be applied across input/output sections, graphical interface, and source code
  - Time-critical source code will be optimized using profilers, different algorithms, and data structures
  - Standard components will be used where possible (i.e., commercial numerical libraries in place of heuristic or experimental solutions)
  - Components will be broken out into individually exercisable pieces and automated test suites will be built
  - Exception propagation and recovery will be introduced into critical sections
  - Graphical interface will be reworked for better stability and usability
  - Extensive user and developer documentation will be developed
  - Interoperability with industry standard applications will be increased

Beyond Phase I

- VTB will move from academic software to a commercially viable, optimized, and well documented suite of tools
- VXE (visualization environment) will become Vizor, maturing in the same ways as VTB
- Rich component database will be created using validated models from marine system suppliers
- Disciplines from all of HM&E will be represented
- Use at NGSS, helping to solve real problems
- All input welcome…

Outline

- Objectives
- Participants
- History
- Concepts
- Approach
- Status

Status

- Currently few months into S3D project
- Expect to deliver Phase 1 demonstration in Dec 2004